

UNITED STATES PATENT APPLICATION

FOR

**METHOD AND APPARATUS FOR STORING COMPOSITE DATA STREAMS**

**Inventors:**

**R. Hugo Patterson  
Allan Bricker  
Richard Johnsson**

**Prepared by:**

**BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP  
12400 Wilshire Boulevard  
Seventh Floor  
Los Angeles, California 90025  
(408) 720-8300**

# **METHOD AND APPARATUS FOR STORING COMPOSITE DATA STREAMS**

## **BACKGROUND OF THE INVENTION**

### **Field of the Invention**

[0001] The invention relates to the field of data storage. More specifically, the invention relates to storing composite data streams.

### **Background of the Invention**

[0002] The amount of data to be stored continues to grow. In particular, the size of the applications and the data generated there from is increasing. Moreover, systems/users are backing up multiple copies of a given set of data to maintain multiple versions. For example, snapshots of a given database stored in a server are copied and stored over time, thereby allowing a given version/snapshot of a set of data to be restored.

[0003] There are existing backup systems that use what are called composite data streams. Figure 1 is a diagram of composite data streams generated for storage as a backup according to the prior art. In Figure 1, at a first time a constituent user data stream 103 is being backed up. The contents of the constituent user data stream is conceptually illustrated as a series of letters “APKLZATUALMNOAKAPLY...” These letters may represent a variety of different levels of granularity of data and/or boundaries, including fixed sized chunks regardless of file boundaries, different files, fixed sized chunks within file boundaries etc. The constituent user data stream 103 is combined (e.g., multiplexed) with a constituent administrative data stream 104 to form a composite data stream 101 (e.g., a first snapshot) for backup storage. In other words, the constituent user data stream 103 is broken into data stream blocks that are interleaved with data stream blocks of the constituent administrative data stream 104 (e.g., tape markers, time stamps, hashes, error correction data, etc.).

[0004] A dashed line in the middle of Figure 1 separates a second backup operation performed at a later time (a second time). In particular, at this later time the user data has been modified, and thus a constituent user data stream 105 is formed. The constituent user data stream 105 is conceptually illustrated as “APKLZAUALMNOAKAPLY...” Thus, the difference between the constituent user data streams 103 and 105 is that the “T” has been removed from the constituent user data stream 105. The constituent user data stream 105 is combined with a constituent administrative data stream 106 to form a composite data stream 109 (e.g., a second snapshot) for backup storage. Since the constituent user data stream 105 is different from the constituent user data stream 103, the resulting composite data stream 101 is different from the composite data stream 109 (even if the constituent administrative data streams 104 and 106 are the same); In particular, at least certain of the data stream blocks of the constituent user data stream 103 in the composite data stream 101 contain different data than the data stream blocks of the constituent user data stream 105 in the composite data stream 109. Similarly, if the constituent administrative data streams 104 and 106 were different, the resulting composite data streams 101 and 109 would be different even if the user data (the constituent user data streams 103 and 105) had remained the same.

[0005] To provide an exemplary use of composite data streams, backup clients residing on different computers of a local area network may be provided and/or collect data to be backed up on their respective computers. This data to be backed up may or may not be in the form of a composite data stream as a result of the application(s) which created it. These backup clients may each transmit (e.g., over a network) data streams (e.g., constituent user data streams, which themselves may be composite data streams) to a backup server that forms composite data streams (e.g., by combining a constituent user data stream with one or more other constituent user data streams and/or an administrative

data stream). It should be thus understood that there may be multiple layers of composite data streams. The backup server periodically transmits (e.g., directly or over a network) these composite data streams to a storage server (e.g., a network file server, a tape library emulator server, etc.) for storage, as well as maintains a catalog of the backups it is managing and what it has stored therein. Although forming composite data streams is common, different backup systems structure composite data streams differently (e.g., certain backup systems use fixed length blocks of user data separated by administrative data blocks; other backup systems punctuate variable length user files with administrative data; etc.). The structure of the composite data streams is often chosen as a result of the type of storage media used by the storage server (e.g., tape, magnetic disk, optical disc, etc.). These storage servers typically store each of these entire composite data streams as separate files. Thus, there is an existing base of backup software used in backup clients and backup servers that provide composite data streams to storage servers.

[0006] Typically, much of the data across different snapshots remains the same (e.g., there is little difference between the constituent user data streams 103 and 105). For example, if the data is backed up for a given user on a daily basis and such user is updating only one of the number of files on a given day, the data in this file is the only data that has been modified. As a result, storage servers that store entire composite data streams are relatively inefficient in that they store large amounts of redundant data.

[0007] There are some backup systems that allow for the sharing of data across a number of different snapshots/versions to reduce the amount of data being stored. Such backup systems are referred to as segment reuse backup systems. Segment reuse backup systems typically operate by breaking up the data for each snapshot into segments. The segments of a current snapshot are compared to the segments of a previous snapshot to determine if there are matching segments. For any segments that match, only a pointer to the segment

of the previous snapshot need to be stored to backup that segment from the current snapshot. In this manner, the efficiency of the backup system is improved by reducing the storage of redundant data.

## **BRIEF SUMMARY OF THE INVENTION**

[0008] A method and apparatus for storing composite data streams is described.

According to one embodiment of the invention, a composite data stream is stored so that it may be restored. The storing of the composite data stream includes decomposing the composite data stream into a plurality of constituent data streams, segmenting at least one of the plurality of constituent data streams, and discarding those of the segments resulting from the segmenting which are determined to have been stored previously.

[0009] These and other aspects of the present invention will be better described with reference to the Detailed Description and the accompanying Figures.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0010] The invention may best be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

[0011] Figure 1 is a diagram of composite data streams generated for storage as a backup according to the prior art.

[0012] Figure 2A is an exemplary diagram illustrating decomposing a composite data stream according to one embodiment of the invention.

[0013] Figure 2B is an exemplary diagram illustrating recomposing a composite data stream from files according to one embodiment of the invention.

[0014] Figure 3 is an exemplary diagram of a composite data stream decomposer/recomposer segment reusing storage server according to one embodiment of the invention.

[0015] Figure 4 is a flowchart for decomposing a composite data stream according to one embodiment of the invention.

[0016] Figure 5 is a flowchart for recomposing a composite data stream from constituent data streams according to one embodiment of the invention.

[0017] Figure 6 is an exemplary diagram of a composite data stream map file according to one embodiment of the invention.

[0018] Figure 7A illustrates application of segment reuse to composite data streams.

[0019] Figure 7B illustrates an example of decomposing composite data streams prior to segmentation according to one embodiment of the invention.

## **DETAILED DESCRIPTION OF THE INVENTION**

[0020] In the following description, numerous specific details are set forth to provide a thorough understanding of the invention. However, it is understood that the invention may be practiced without these specific details. In other instances, well-known circuits, structures, standards, and techniques have not been shown in detail in order not to obscure the invention.

[0021] Figure 7A illustrates application of segment reuse to composite data streams. In Figure 7A, segments are started on each occurrence of the letter A for purposes of illustration; in embodiments of the invention, any number of techniques can be used to anchor segments in a data stream – e.g., a repeated pattern, a repeated hash pattern, etc. Thus, Figure 7A illustrates that the composite data stream 101 has been divided into segments 707A-707E (707A=APKL, an administrative data block, and Z; 707B=ATU and an administrative data block; 707C=ALMN, an administrative data block, and O; 707D=AK; and 707E=A, an administrative data block, and PLY), while the composite data stream 109 has been divided into segments 708A-708E (708A=APKL, an administrative data block, and Z; 708B=AU; 708C=A, a first administrative data block, LMNO, and a second administrative data block; 708D=AK; and 708E=AP, an administrative data block, and LY). In addition, Figure 7A illustrates the comparison of segments 707A-E to the segments 708A-E with the same letter. If the constituent administrative data streams 104 and 106 are the same, then the segments 708B, 708C, and 708E do not match and must be separately stored. If constituent administrative data streams 104 and 106 are not the same, then the segment 708A also does not match and must also be separately stored. (It should be understood that even in the alternative case where the user data did not change but the constituent administrative data streams did, then four segments would not find a match because segments 707A, 707B, 707C, and



707E each include an administrative data block). In either case, a number of segments need to be stored by the storage server applying segment reuse even though the change to the data was relatively minor. This results in relatively low compression efficiency and consumes resources, especially storage, to compress and store two versions of the same data.

[0022] Figure 7B illustrates an example of decomposing composite data streams prior to segmentation according to one embodiment of the invention. Figure 7B illustrates the composite data streams 101 and 109 being provided at different times. Each of the composite data streams 101 and 109 is decomposed into its constituent data streams prior to being segmented. Figure 7B illustrates that the constituent user data stream 103 has been divided into segments 717A-717E (717A=APKLZ; 717B=ATU; 717C=ALMNO; 717D=AK; and 717E=APLY), while the constituent user data stream 105 has been divided into segments 718A-718E (718A=APKLZ; 718B=AU; 718C=ALMNO; 718D=AK; and 718E=APLY). In addition, Figure 7B illustrates the comparison of segments 717A-E to the segments 718A-E with the same letter. Regardless of whether the constituent administrative data streams 104 and 106 are the same, only the segment 718B of the segments 718 does not match and must be separately stored (that is, segments 718A, 718C, 718D, and 718E need not be stored – only a reference to segments 717A, 717C, 717D, and 717E need be stored). (It should be understood that a similar effect applies to the constituent administrative data streams). This results in relatively higher compression efficiency and consumes fewer resources, especially storage.

[0023] Figures 2A – 2B are exemplary diagrams illustrating a data stream decomposer/recomposer according to one embodiment of the invention. Figure 2A is an exemplary diagram illustrating decomposing a composite data stream according to one embodiment of the invention. In Figure 2A, a composite data stream

decomposer/recomposer 207 receives a composite data stream 201. The composite data stream 201 includes interleaved data stream blocks from a constituent data stream A (see data stream blocks 203) and a constituent data stream B (see data stream blocks 205). A data stream configuration file 215 provides information to the composite data stream decomposer/recomposer 207 regarding the structure used by the backup system that created the composite data stream 201. The composite data stream decomposer/recomposer 207 uses the information provided by the composite data stream configuration file 215 to decompose the composite data stream 201. The composite data stream configuration file 215 may be a file created by an administrator, a file created and received remotely, a default configuration file, etc. In an alternative embodiment of the invention, the composite data stream decomposer/recomposer 207 processes a composite data stream without a composite data stream configuration file. For example, the composite data stream decomposer/recomposer 207 determines from the composite data stream itself the structure of the composite data stream (e.g., a certain number of bits are stored and analyzed for a certain bit pattern that indicates the structure of the composite data stream, initialization data in the composite data stream indicates the structure of the composite data stream, etc.).

[0024] The composite data stream decomposer/recomposer 207 decomposes the composite data stream 201 into the constituent data stream A 211 and the constituent data stream B 213. The composite data stream decomposer/recomposer 207 also generates a composite data stream map 209. The composite data stream map 209 indicates how the composite data stream was decomposed into the constituent data streams 211 and 213.

[0025] Figure 2B is an exemplary diagram illustrating recomposing a composite data stream according to one embodiment of the invention. In Figure 2B, the composite data stream decomposer/recomposer 207 recomposes the composite data stream 201 from the

constituent data stream A 211 and the constituent data stream B 213 in accordance with the composite data stream map 209. Thus, the composite data stream 201 includes the interleaved data stream blocks 203 and 205. An example of a composite data stream map file will be described later.

[0026] The composite data stream decomposer/recomposer 207 illustrated in Figures 2A – 2B can be implemented as software, hardware (e.g., an application specific integrated circuit), or a combination of hardware and software.

[0027] It should be understood that while embodiments of the invention are described herein (e.g., see Figures 7B, 2A, 2B, 4, 5, 6) with reference to an exemplary composite data stream made up of two constituent data streams (and often a constituent user data stream and a constituent administrative data stream), the invention is not limited to such composite data streams. Rather, the invention is applicable to a composite data stream formed by combining any number of different constituent data streams (e.g., one or more constituent user data streams and zero or more administrative data streams). In addition, the invention is applicable to composite data streams that have multiple layers (while a given composite data stream is made up of its constituent data streams, one or more of these constituent data streams may themselves be composite data streams; Thus, the term constituent data stream refers to a data stream (be it a composite data stream itself or not) that is combined with other data streams to form a composite data stream). It should be understood that in the case of a composite data stream with multiple layers, recursively decomposing the input composite data stream (either completely (into the smallest divisible data streams), partially (down a certain number of levels), selectively (certain constituent data streams are recursively decomposed further than others), etc.) is within the scope of the invention.

[0028] Figure 3 is an exemplary diagram of a composite data stream

decomposer/recomposer segment reusing storage server according to one embodiment of the invention. The storage server 311 (e.g., a network file server, a tape library emulator server, etc.) includes a set of one or more interface agents 309, a composite data stream decomposer/recomposer 313, a segment reuse storage system 317, a map file storage 315, and (optionally) a constituent data stream regenerator 320. In one embodiment of the invention, the map file storage 315 and the segment reuse storage system 317 are a single storage unit, whereas in alternative embodiment of the invention the map file storage 315 and the segment reuse storage system 317 are multiple storage units.

[0029] Composite data streams may be communicated between one or more backup servers and the storage server 311 in a variety of ways - e.g., directly or over a network (e.g., LAN, SAN, WAN, etc.) using a link (e.g., wirelessly, Ethernet, fiber channel, FDDI, ATM, SCSI, etc.) and a protocol (e.g., TCP/IP, NFS, CIFS, NDMP, SCSI, etc.) that may or may not be layered.

[0030] The interface agent 309 communicates composite data streams with one or more backup servers. Incoming composite data streams are sent to the composite data stream decomposer/recomposer 313. In an alternative embodiment of the invention, the interface agent 309 and the composite data stream decomposer/recomposer 313 are implemented as a single module. The composite data stream decomposer/recomposer 313 decomposes (or demultiplexes) composite data streams into constituent data streams and creates map files to aid in their recomposition. The constituent data streams are stored by the segment reuse storage system 317. The map files generated by the composite data stream decomposer/recomposer 313 are stored in the map file storage 315. The optional administrative data regenerator 320 regenerates administrative data using an algorithm as described later herein.

[0031] Although Figure 3 illustrates the storage server as including the map file storage 315 and the segment reuse storage system 317, alternative embodiments of the invention implement the segment reuse storage system 317 and/or the map file storage 315 separately from the storage server 311. For example, the composite data stream decomposer/recomposer 313 may decompose a composite data stream into constituent data stream files and transmits them over an Ethernet to a segment reuse storage farm for storage. Alternatively, the storage server 311 includes the map file storage 315 and the segment reuse storage system 317 as illustrated and is also networked to a storage farm.

[0032] While the backup clients and servers may be developed in conjunction with the storage server 311, the storage server 311 may be used with an existing composite data stream backup system, including the existing base of software used in backup clients and backup servers. In particular, the storage server's decomposing at backup and recomposing at restore provides the input and output (the composite data stream) expected by the existing base of composite data stream software, while at the same time allowing for more efficient storage. In addition, through the use of configuration files and/or composite data stream preprocessing, the storage server 311 may be made compatible with multiple different backup systems and/or different versions of backup system(s).

[0033] The storage server described above includes memories, processors, and/or ASICs. Such memories include a machine-readable medium on which is stored a set of instructions (i.e., software) embodying any one, or all, of the methodologies described herein. Software can reside, completely or at least partially, within this memory and/or within the processor and/or ASICs. For the purpose of this specification, the term "machine-readable medium" shall be taken to include any mechanism that provides (i.e., stores and/or transmits) information in a form readable by a machine (e.g., a computer).

For example, a machine-readable medium includes read only memory ("ROM"), random access memory ("RAM"), magnetic disk storage media, optical storage media, flash memory devices, electrical, optical, acoustical, or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.), etc.

[0034] Figure 4 is a flowchart for decomposing a composite data stream according to one embodiment of the invention. At block 401, the structure of a composite data stream is established. As previously stated, the structure of a composite data stream may be established with a configuration file created by an administrator or received over a network, a default configuration file and/or settings, composite data stream preprocessing, information within the composite data stream, etc. At block 403, the composite data stream is received, for example, by the composite data stream decomposer/recomposer 313 of Figure 3 or 207 of Figures 2A and 2B. At block 405, the composite data stream decomposer/recomposer determines if the structure of the composite data stream indicates that the composite data stream includes any administrative data that the system will restore by regenerating it (in other words, the system will restore that administrative data using an algorithm, as opposed to by accessing a copy that was stored). For example, if an embodiment of the invention implements the algorithm for generating the administrative data (the optional administrative data regenerator 320) and has all of the necessary inputs, the administrative data can be restored by regenerating it (determining it on the fly/dynamically). In contrast, if an embodiment of the invention does not implement the algorithm (e.g., it is unknown, unavailable, etc.), does not have all of the necessary inputs (e.g., one or more is unknown, unavailable, etc.), and/or the data is not of a nature that can be regenerated, the administrative data needs to be stored. If the composite data stream does include any administrative data that the system will restore by regenerating it, then control flows to block 407. However, if the composite data stream

does not include any administrative data that will be regenerated , then control flows to block 409.

[0035] At block 407, the composite data stream decomposer/recomposer decomposes the composite data stream into its constituent data streams and generates a map file, but discards any administrative data that the system will restore by regenerating it. For example, in certain embodiments of the invention, tape markers are discarded and not stored in a constituent data stream file(s).

[0036] At block 409, the composite data stream decomposer/recomposer decomposes the composite data stream into its constituent data streams and generates a map file.

[0037] Figure 5 is a flowchart for recomposing a composite data stream from constituent data streams according to one embodiment of the invention. At block 501, a command is received to recompose a composite data stream. The command may be from a user, from a backup server received over a network, from an agent that submits a command periodically, etc. At block 503, the composite data stream decomposer/recomposer retrieves the composite data stream map file (including any structure information) for the requested composite data stream. At block 505, the composite data stream decomposer/recomposer determines if the structure information indicates that any administrative data needs to be regenerated because it was not stored as part of the backup. If the structure information indicates that administrative data is to be regenerated, then control flows to block 507. Otherwise, control flows to block 509.

[0038] At block 507, the composite data stream decomposer/recomposer recomposes the composite data stream from constituent data streams according to the composite data stream map file, while regenerating (by a technique other than retrieval from the backup storage (e.g., calculated)) and inserting administrative data.

[0039] At block 509, the composite data stream decomposer/recomposer recomposes the composite data stream from its constituent data streams according to the map file.

[0040] Decomposing and recomposing data streams as described in Figures 4 and 5 further reduces the storage necessary for backing up composite data streams. In particular, data streams, or parts thereof, that can be regenerated, such as administrative data (e.g., tape markers), can be discarded and restored without consuming storage space. In addition, resources are not spent compressing and storing such data streams.

[0041] Figure 6 is an exemplary diagram of a composite data stream map file according to one embodiment of the invention. A composite data stream 601 includes interleaved data stream blocks of a constituent data stream A 603 and a constituent data stream B 605. The composite data stream 601 is sectioned into a section 1 607 and a section 2 609. The section 1 607 and the section 2 609 logically illustrate sectioning of the composite data stream 601 for storage as files. Although section 1 607 and section 2 609 are the same size in Figure 6, composite data streams may be sectioned into varying sizes in accordance with the described invention.

[0042] Section 1 607 and section 2 609 each include data from both constituent data streams 603 and 605.

[0043] Figure 6 also illustrates a composite data stream map. The composite data stream map includes a composite data stream map header 611 and a composite data stream map block for each section (see composite data stream map block 613 for section 1 607).

While in one embodiment of the invention a data stream map header and corresponding data stream map blocks are a single file, in alternative embodiments of the invention they are separate files. In Figure 6, the composite data stream map header 611 includes a composite data stream identifier field, a total number of constituent data streams field, and a constituent data stream identifier field for each constituent data stream of the



composite data stream. Table 1 illustrates example data in the composite data stream map header 611.

1A56YZCLIENTX (Identifier for the composite data stream 601)
2
1A56YZCLIENTX A
1A56YZCLIENTX B

Table 1: Example Data in a Composite Data Stream Map Header

[0044] The first field in table 1 is an example identifier that can be used to identify a composite data stream; the second field indicates that there are two constituent data streams; and the third and fourth fields contain an example identifier that can be use to identify constituent data streams. Various techniques can be employed to assign identifiers to composite and constituent data streams. Alternatively, identifiers assigned by the source of the composite and/or constituent data streams can be used to differentiate between data streams. For example, in one embodiment of the invention, an identifier is the composite data stream identifier used by the source of the composite data stream and an identifier that identifies the source. .

[0045] The composite data stream map block 613 illustrated in Figure 6 includes a composite offset field, a constituent data stream offset field for each constituent data stream, and a list of composite data stream descriptors. Each composite data stream descriptor includes an identifier field for the constituent data stream corresponding to the next data stream block of the composite data stream and a length field for the length of that data stream block. The composite offset field indicates the offset in the composite data stream of the data specified by the first composite data stream descriptor in the composite data stream map block. Each descriptor indicates, in order, how much of which constituent data stream to take next to recompose the composite data stream. Each constituent data stream offset field indicates the offset in the constituent data stream of the first data specified by the first composite data stream descriptor in the map block

which includes the identifier for the corresponding constituent stream. Table 2 provides an example of data in a composite data stream map block.

600	
400	
200	
A	100
B	50
A	100
B	50
...	...

Table 2: Example Data in a Composite Data Stream Map Block

[0046] Assuming the values indicated in table 2 are in kilobytes, table 2 shows that 600 kilobytes of the composite data stream precedes the data described by the map block, 400 kilobytes of the constituent data stream A precedes the data described by the map block, and 200 kilobytes of the constituent data stream B precedes the data described by the map block. The next data in the composite data stream is 100 kilobytes from constituent stream B, followed by 50 kilobytes from constituent data stream A, and so on.

[0047] What is represented and the size of the map blocks depends on the implementation (e.g., a fixed number of bytes from the composite data stream are represented by each map block, a fixed number of data stream blocks are represented by each map block, etc.). While in one embodiment the composite data stream is sectioned, alternative embodiments of the invention do not section the composite data stream. In alternative embodiments of the invention, the map files also include indexing into data structures (e.g., trees, hash tables, etc.) that store the constituent data stream files. In another embodiment of the invention, the map files include indexing that is used to recompose the composite data stream without offset fields.

[0048] In addition to increasing the compression efficiency and reducing storage consumption, decomposing composite data streams into its constituent data streams enables selective retrieval of data from storage. For example, instead of restoring an

entire composite data stream, a single constituent data stream can be selected and restored to a requesting entity.

Alternative Embodiments

[0049] While the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. For instance, while the flow diagrams show a particular order of operations performed by certain embodiments of the invention, it should be understood that such order is exemplary (e.g., alternative embodiments may perform the operations in a different order, combine certain operations, overlap certain operations, etc.).

[0050] Thus, the method and apparatus of the invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting on the invention.